

REPLY TO EXAMINER'S ANSWER

First Ground of Rejection:

Claims 1, 12 and 20 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Aldred in view of Raynak. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 12 and 20:

Regarding claim 1, Appellants have argued that, contrary to the Examiner's assertion, Aldred in view of the Raynak does not teach or suggest a first application launching a second application, where the launching of the second application includes the first application passing an event port number and a command port number to the second application. The Examiner cites various portions of Aldred (column 5, lines 51-63; column 6, lines 39-49; column 7, lines 33-62; column 12, lines 57-61; and column 36, lines 3-52) that describe Aldred's channels and share sets. However, none of these cited passages describes passing event port numbers and command port numbers to an application as part of launching that application.

Instead, Aldred specifically teaches the use of support system software together with call manager applications to establish, configure, and manage communication channels between applications, especially between applications executing on different hardware nodes (Abstract; column 1, lines 52-60). Aldred teaches that groups of applications communicate by participating in named sharing sets. Aldred's call managers coordinate, monitor and manage the various share sets of applications. Aldred also teaches a support system and a software API through which applications interact with the call managers. Aldred's API includes functions for initiating and configuring communication between shared applications via channels and signals. However, none of the teachings of Aldred describe a first application launching a second application, where

the launching of the second application includes the first application passing an event port number and a command port number to the second application.

The Examiner also relies on Raynak to teach one application launching another application and passing communication parameters as part of launching the launched application. Raynak teaches a method for transferring a network connection to an external host to a second application. Raynak teaches that an Information Manager (IM) application may launch a secondary application and pass a communication port number (as well as a baud rate) as command line parameters when executing the secondary application. The Examiner cites FIG. 4, column 1, lines 11-24 and column 6, lines 32-57 of Raynak. However, Raynak does not teach or suggest passing a command port number and an event port number. Instead, Raynak only teaches passing a communication port ID (e.g. COM1, COM2, etc) and a baud rate. Thus, the Examiner's combination of Aldred and Raynak does not teach or suggest that the launching of the second application includes the first application passing an event port number and a command port number to the second application.

It is the Examiner's contention that an application in Aldred's system *could* pass an event port number and command port number when launching a second application. However, as noted above, Aldred does not teach or suggest such functionality. The Examiner relies on the fact that Aldred teaches the use of a `launch_app` function that includes several parameters and that Aldred teaches that a channel is defined by the first, sending application. The Examiner cites column 11, lines 27-39 and column 29, lines 8-19, where Aldred describes launching applications and refers to Aldred's "`launch_app`" API command. However, Aldred's `launch_app` function is used by applications to interact with, and request support services from, Aldred's call managers (*see*, Aldred, column 4, lines 27-43). Thus, an application wishing to launch another application uses the `launch_app` function to communicate the request to a call manager. The call manager forwards the request to a call manager executing on the appropriate node of Aldred's system. The second call manager may then launch the requested application (Aldred, column 5, lines 51-63). The fact that Aldred includes a mechanism to launch applications

does not teach or suggest passing an event port number and a command port number as part of launching an application. **Nowhere does Aldred describe passing event port numbers and command port numbers as part of launching an application via the launch_app API function.** In contrast, Aldred teaches that an application may issue a launch_app command and may be returned a limited use handle to the launched application that is “only valid in very restricted circumstances *until the launched application has registered with the support system*” (emphasis added, column 11, lines 34-36).

The manner and method of initializing and configuring communication channels between applications described in Aldred does not include passing event port numbers and command port numbers as part of launching an application. To the contrary, Aldred explains the benefits of using the support system software when establishing and configuring communications channels. For example, relying upon call managers and the support system software allows applications to be “aware” of, and to use, Aldred’s system while avoiding the need to be involved in “call set-up or tear-down.” Aldred teaches the benefit of providing clear separation of call management and application programming (Aldred, column 26, lines 62-67). Thus, Aldred specifically does not describe a first application launching a second application, where the launching of the second application includes the first application passing an event port number and a command port number to the second application.

Aldred states, “in order for an application instance to be allowed to communicate with the system, it must identify itself by issuing a register_app call” (column 35, lines 48-67). Aldred also teaches, “it is up to the launched application to use [the] register_app [function] to fully identify itself to the system” (column 36, lines 21-55). Aldred describes that adding a port to a channel includes a request from one application, which is sent *via the support system* as an unsolicited event to a second application, and a confirmation (or error) response routed back to the first application as a confirm event (Aldred, column 24, lines 39-51). Thus, Aldred teaches the use of his support system function calls to send ports numbers to another application. **Thus Aldred very clearly**

teaches that ports, and therefore event port numbers and command port numbers, are only configured after an application has registered with the support system. Nowhere does Aldred describe passing event port numbers and command port numbers to an application *as part of launching that application*.

Additionally, one of the benefits of Aldred's share sets, call managers and support system software is that data may be communicated across heterogeneous networks using passive nodes to route data between an application on one node and another application on another node (Aldred, column 2, lines 19-50; column 5, lines 41-50; column 19, lines 24-48). If, as the Examiner contends, a launching application sends event and command port numbers to a launched application as parameters to Aldred's launch_app function, Aldred's system would no longer be able to use intermediate nodes to route data, since the intermediate nodes would not receive the event and command port numbers since the parameters to Aldred's launch_app function are not distributed to the intermediate nodes.

The Examiner has stated that he "believes it is reasonable to suggest that the parameters passed in Aldred's launch_app function would be the channel characteristics needed for launching and launched applications to communicate". However, the Examiner's belief is completely unsupported by the actual teachings of the cited art, and can thus only be based on hindsight knowledge of Appellants' disclosure. In fact, **Aldred teaches away** from one application passing event port numbers and command port numbers as part of launching another application. As described above, Aldred's system already includes a very specific mechanism to initiate and configure ports and channels between applications that specifically does not include passing event port numbers and command port numbers as part of launching applications. Aldred clearly teaches the benefits of an application first registering with a call manager and joining a share set before initializing or configuring channels and ports. Rather than providing any motivation to modify Aldred's system to pass event port numbers and command port numbers as part of one application launching another application, **Aldred teaches away** from one application launching another application *and passing event port numbers and command port numbers as part of launching the other application*. Furthermore, it

would not make sense to modify Aldred to bypass the share sets that are central to Aldred's system by passing event port numbers and command port numbers as part of launching applications. Such a modification would not only be contrary to Aldred's specific teachings, it would remove the specific benefits of Aldred's sharing sets, call managers, and support system software.

The Examiner, in the Advisory Action and Examiner's Answer, responds to this argument by repeating the assertion that Aldred's launch_app function *might* include command and event port numbers and again citing column 36, lines 24-53 of Aldred. This is pure unsupported hindsight speculation by the Examiner. As noted above, this passage of Aldred fails to mention sending command and/or event port numbers as part of one application launching another application. Moreover, the cited passage describes the "parameters" referred to by the Examiner as "a user specified string that is given to the launched application". The Examiner appears to be interpreting the "user specified string" as including command and event port numbers. However, nowhere does Aldred mention that a user of his system specifies command and event port numbers. Instead, as noted above, Aldred system includes a specific set of APIs allowing applications to setup command and event ports, including specifying port numbers. **Without some specific teaching by Aldred, the Examiner's contention that rather than use the specific support system function calls to initialize command and event port numbers an application in Aldred's system would require the user to specify command and event port numbers is clearly incorrect.**

In the Examiner's Answer, the Examiner asserts that Appellants' discussion of Aldred's mechanism to initiate and configure ports and channels between applications, "shifts the focus away from the limitations set forth in the claim." However, Appellants' discussions regarding Aldred's use of support system software to initialize and configure ports and channels are in rebuttal to the Examiner's (erroneous) contention that "the parameters passed in Aldred's launch_app function would be the channel characteristics needed for launching and launched applications to communicate". Thus, the Examiner contends that an application in Aldred's system needs to pass event and command ports

numbers via Aldred's launch_app function in order to communicate. Appellants' respond, as noted above, that the Examiner's contention is incorrect. Appellants discuss the specific mechanism that Aldred teaches for allowing applications to communicate which does not involved passing event and command port numbers via a user specified string parameter of the launch_app function.

The Examiner also asserts, in the Final Office Action, the Advisory Action, and the Examiner's Answer, that "the sending application is responsible for defining the channel characteristics", and that "modification of Aldred to include passing an channel and port information between applications does not change the principle of his invention". The Examiner cites column 1, lines 61-65. However, the Examiner has clearly mischaracterized Aldred's meaning regarding "the sending application is responsible for defining the channel characteristics" by stating instead "the sending application is responsible for establishing the channel between applications". *Defining the channel characteristics* and *establishing the channel* are two different actions. In fact, Aldred describes these characteristics in column 12, lines 57-64 as: "target application handle, channel set type and identifier, data class, maximum buffer size, user exit, node handle, quality of service, connect type, port event handler, user information". In Aldred's system, the sending application must provide the support system with this information in channel creation; but *it does not establish the channel itself*. Appellants also note that in creating the channel between two programs, i.e., the launched and launching programs, a target application handle and a node handle are required by the support system. The Examiner speculates (erroneously) that "parameters passed in Aldred's launch_app function would be the channel characteristics needed for the launching and launched applications to communicate". However, the Examiner has not cited any portion of the art to support such a suggestion. As noted above, a target application handle and a node handle are required for the channel creation, and as stated at column 11, lines 27-39, are not available until *after* the application has been launched. Furthermore, the node handle is specified by the return data, which is returned *after the application has registered* with the support system (column 11, lines 29-31 and 36-39).

Appellants have further argued that, contrary to the Examiner's assertion in his Response to Arguments, the limited use handle does not disclose "that a channel has been already established between the applications". In fact, Aldred defines how the handle is implemented in column 36, lines 45-48: "This function [launch_app] is used to ask the system to start another program instance. If the new application is started successfully then its instance handle is inserted in the target_application and returned to the calling application". Aldred has already defined the target_application as a pointer used by the system. Therefore, Aldred's system changing the value of a particular pointer, where that changing of value is communicated within the support system and not directly between the two programs, does not teach that a channel has been already established between the applications. And, as noted above, the newly launched application is not "allowed to communicate with the system" without registration. Furthermore, this communication with the system is required for the system to create a channel as in the API call function add_channel or create_channel (column 29).

Additionally, the Examiner asserts, in the Advisory Action, "Aldred discloses that the register function merely allows the support system to be aware of the application", citing column 36, lines 9-16. **However, the cited passage makes no such statement.** Instead, the cited passage describes how if no call manager currently exists the issuer of the register_app call either becomes the call manager or terminates. The Examiner has misrepresented the true teachings of Aldred.

Furthermore, the Examiner has stated that nowhere does Aldred declare that ports are only configured after an application has registered with the support system. Apparently the Examiner has overlooked the fact that Aldred specifically discloses that: "[i]n order for an application instance to be allowed to communicate with the system, it must identify itself by issuing an register_app call" and that, "[t]his call [register_app] must be issued prior to any other calls from this instance, otherwise the calls will fail." Additionally, Aldred teaches, "it is up to the launched application to use an register_app to fully identify itself to the system" (column 35, lines 47-67). Clearly, Aldred's registration allows the newly launched application to interact with the system and is not

simply limited to “allow other applications to know that it has been launched”, as the Examiner contends.

In the Examiner’s Answer, the Examiner argues that Appellants discussion of Aldred’s teachings regarding the support system software, call managers, establishing channels and configuring ports “have no bearing on the limitation of whether port numbers are passed to an application at the time it is launched.” The Examiner has misunderstood Appellants’ argument. Appellants’ argument is not that claim 1 recites establishing channels and configuring ports. Rather, Appellants’ argument is that Aldred teaches a support system that includes establishing channels, configuring ports and *passing port numbers* between applications. As noted above, Aldred teaches the use of his support system function calls to send ports numbers to another application. (Aldred, column 24, lines 39-51). Appellants’ discussions of Aldred’s support system software, call managers and establishing channels and configuring ports are to specifically rebut the Examiner’s arguments and to show that Aldred’s teaching does not support the Examiner’s contention that an application in Aldred’s system would pass an event port number and a command port number via a “user supplied string” when launching another application.

In the Examiner’s Answer, the Examiner also states, “Aldred merely discloses several functions that enable specifying port information but in no way does he limit other times when the port information can be submitted between the applications.” Even if Aldred’s teachings do not limit other times when the port information can be submitted between the applications, that does not mean that Aldred actually teaches the specific limitations of Appellants’ claimed invention. Aldred clearly fails to teach or suggest that applications may pass port information in any other manner than via the support system software. The Examiner has not cited any teaching of Aldred that suggest that applications pass port information via the launch_app function. Instead, the Examiner’s position is that since port information *could* (in the Examiner’s opinion) be passed via the user string parameter of the launch_app function, Aldred somehow suggests the specific

limitation of Appellants' claim. The Examiner position is a blatant hindsight reconstruction of Appellants' claimed invention.

Additionally, as discussed in the M.P.E.P. at 2142, "to support the conclusion that the claimed invention is directed to obvious subject matter, either the references must expressly or impliedly suggest the claimed invention or the examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references". As shown above, Aldred and Raynak, whether considered singly or in combination, fail to teach or suggest a first application launching a second application, where the launching of the second application includes the first application passing an event port number and a command port number to the second application. The Examiner has failed to provide a convincing line of reasoning as to why one would find Appellants' invention obvious in light of the cited art. It is simply not obvious that an application in Aldred's system would not use Aldred's support system software to pass event and command port numbers, thereby circumventing the benefits of using the support system software, and instead pass event and command port numbers via a *user specified* string as a parameter to the launch_app function. The Examiner's line of reasoning is certainly not convincing in light of the fact that Aldred does not teach or suggest that event and command port numbers can be passed via the user specific string parameter of the launch_app function. In fact, Aldred fails to mention or suggest that event and command port numbers have anything to do with the user specific string parameter of the launch_app function.

Moreover, modifying Aldred's system to include passing an event port number and a command port number to an application as part of launching that application would change the principle of operation of Aldred's system. Aldred's system relies upon applications registering and utilizing both the call managers and the support system software via Aldred's API to properly initiate and configure communications between applications. Bypassing this system to send event port numbers and command port numbers to applications, as part of launching those applications, would bypass Aldred's sharing set concept, which is essential to the operation of his

system, and thus change Aldred's principle of operation. As discussed in M.P.E.P. § 2143.01, a rejection based on a modification that changes the principle of operation of a reference is improper.

In the Examiner's Answer, the Examiner argues that Aldred's teachings regarding the call managers and support system initializing and configuring communications "are completely separate from the limitation of passing port numbers as part of the launching process." However, the Examiner has failed to provide any rebuttal to the fact that modifying Aldred's system to pass event and command port numbers via a user specified string on a launch_app function would clearly bypass Aldred's system of call managers and support software, thus changing the principle of operation of Aldred's system.

Additionally, Appellants have argued that the Examiner's combination of Aldred and Raynak is improper because, as noted above, Aldred teaches away from passing a command port number and an event port number when launching a second application. M.P.E.P. 2144.05.III states that a "prima facie case of obviousness may also be rebutted by showing that the art, in any material respect, teaches away from the claimed invention". See also, *In re Geisler*, 116 F.3d 1465, 1471, 43 USPQ2d 1362, 1366 (Fed. Cir. 1997). Additionally, "[a] reference may be said to teach away when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the applicant." *In re Gurley*, 27 F.3d 551, 553 (Fed. Cir.1994) (*emphasis added*). As noted at M.P.E.P. § 2141.02, paragraph 12, a "prior art reference *must* be considered in its entirety, i.e. as a whole, including portions that would lead away from the claimed invention" (*italics added, underlining in original*). One of ordinary skill in the art, upon reading Aldred's teachings regarding the use (and benefits) of the call manager and joining a share set before initiating or configuring communications channels and ports, as described above, would clearly be discouraged from following Raynak's teachings regarding modifying an initialization file to include command line parameters specifying a communications port and baud rate. Further, one reading the teachings of Raynak regarding the use of command line parameters to specify a communication port

and a baud rate would clearly be led in a direction away from the teaching of Aldred regarding the use of a call manager and a joining a share set. The respective teachings of Aldred and Raynak pertain to very different methodologies that clearly teach away from one another. As such, the Examiner's proposed combination of Aldred and Raynak is improper.

The Examiner also states in the Examiner's Answer, "[t]he mere fact that [Aldred's share set and call manager] may aid an application in initiating and configuring a channel does not teach away from passing port numbers as part of launching an application, which is taught by Raynak." The Examiner is incorrect. Aldred clearly teaches the benefits of an application first registering with a call manager and joining a share set before initializing or configuring channels and ports. It would not make sense to modify Aldred to bypass the share sets that are central to Aldred's system by passing event port numbers and command port numbers as part of launching applications.

Appellants have also argued that the Examiner has failed to provide a proper motivation for combining the teachings of Aldred and Raynak. The Examiner simply states that it would have been obvious without giving any reasons why one would be motivated to do so (Office Action dated October 3, 2005, page 9, lines 10-12). An obviousness rejection that lacks evidence of a suggestion or motivation for one of skill in the art to combine prior art references to produce the claimed invention is defective as hindsight analysis. In addition, the showing of a suggestion, teaching, or motivation to combine prior teachings "must be clear and particular Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence'." *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination.

In the Examiner's Answer, the Examiner states that the motivation to combine Aldred and Raynak "comes from the nature of the problem to be solved." The Examiner

contends that “[b]oth Aldred and Raynak deal with establishing communications for a newly launched application” and that “the reason for modifying Aldred with Raynak is to enable a launched application to be aware of the necessary port information in order to communicate over the network when it is launched.” However, Aldred already has a mechanism, via his call managers and support software for an launched application to be aware of any necessary port information in order to communicate over the network. The Examiner admits, in the Examiner’s Answer, that Aldred “discloses several functions that enable specifying port information but in no way does [Aldred] limit other times when the port information can be submitted between application” (Examiner’s Answer, page 13, lines 17-19). Thus, the Examiner admits that Aldred already teaches a mechanism for specifying event and command port numbers between applications that does not involve passing the port numbers as part of launching an application. In fact, the Examiner also states that Aldred’s teaching regarding the call managers and support system initiating and configuring communications between applications “are entirely separate from the limitation of passing port numbers as part of the launching process” and that “[i]nitiating and configuring communications between applications is not the same feature as a first application launching a second application” (Examiner’s Answer, page 18, lines 11-19). Thus, the Examiner admits that Aldred teaches a mechanism for passing port information that is “entirely separate” from “passing port numbers as part of the launching process”. One skilled in the art desiring to “enable a launched application to be aware of the necessary port information in order to communicate over a network” would simply be motivated to use Aldred’s already established mechanism for specifying port information. One using Aldred’s system would not motivated by Raynak to modify Aldred to pass event and command port numbers via Aldred’s launch_app function, as suggest by the Examiner.

As argued above, the rejection of claim 1 is not supported by Aldred and Raynak, whether considered singly or in combination. Thus, for at least the reasons provided above, Appellants submit that the rejection of claim 1 is unsupported by the cited art and the Examiner has failed to establish a *prima facie* rejection. Similar remarks also apply to claims 12 and 20.

Second Ground of Rejection:

Claims 2-6, 8-11, 13-17 and 19 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Aldred and Raynak, and in further view of Simonoff. Appellants respectfully traverse this rejection for at least the reasons presented above regarding their respective independent claims. Different groups of claims are addressed under their respective subheadings.

Claims 2-4, 8, 9-11, 13-15, and 19:

The rejection of claims 2-4, 8, 9-11, 13-15 and 19 are allowable for at least the reasons presented above regarding their respective independent claims.

Claims 5 and 16:

Regarding claim 5, Appellants have argued that Aldred in view of Simonoff does not teach or suggest passing a function reference value through the command port connection. The Examiner cites column 24, lines 52-61 of Aldred. However, this portion of Aldred is referring to how applications can make asynchronous calls to Aldred's support system software API by including a reference identifier allowing the application to issue command to obtain the status of, or to cancel, an asynchronous API call. The cited passage does not describe passing a function reference value through a command port connection, as suggested by the Examiner. The reference identifier mentioned in the cited passage is not a function reference value that is sent through a command port connection. Instead, it is merely an identifier to allow a calling application to check on the status of an asynchronous API call.

The Examiner asserts that Aldred's reference identifier passes through the command port. However, in Aldred, the status of an API call is handled between the application issuing the call and the support system, and not between the launching and launched applications. The support system is only inherent in Aldred's system, and

therefore, when communication occurs between only an application and the support system, Aldred's command port does not equate to that of the instant application. The Examiner does not refer to Simonoff in the rejection of claim 5. Aldred does not disclose passing a function reference value *through a command port*. Nor does Simonoff overcome Aldred's failure to teach or suggest passing a function reference value through a command port connection.

In the Examiner's Answer the Examiner repeats his contention that Aldred teaches passing a function reference value through the command port connection and states that nothing in the cited section (column 24, lines 52-61) mentions a support system. However, **the Examiner is incorrect**. The cited section is specifically directed toward, the programming consideration regarding "program calls to the API". Aldred's API is specifically the support system software which Appellants have discussed. Aldred specifically states that "application programming interface 20 allows applications 18 to run support services" and refers to "API 20" and "the API" when referring to the programming interface to the support services and support software (Aldred, column 4, lines 27-54).

Furthermore, the Examiner's contention that "[s]ince Aldred discloses utilizing command ports only for passing back of data between applications, it would have been obvious to one of ordinary skill in the art that these reference identifiers, that refer to calls, would be passed through the command port" is incorrect. The reference identifiers relied on by the Examiner are not passed via the command port in Aldred, but instead are used to allow a calling application to check on the status of an asynchronous API call.

Thus, the rejection of claim 5 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks apply to claim 16.

Claim 6:

Regarding claim 6, Appellants have argued that Aldred and Raynak, in further view of Simonoff, does not teach or suggest passing a function parameter through the command port connection. The Examiner cites column 24, lines 39-42 of Aldred. However, this passage of Aldred describes how Aldred's system handles an application's request for a service. Specifically, Aldred teaches that an application requests a service and supplies the appropriate parameters. Another application, supplying the service, is made aware of the request through an "unsolicited event which appears as an indication" to the second application. The response from the second application is routing back to the requesting application as a "confirm event." The cited passage does not mention passing a function parameter through a command port connection, but instead teaches the use of unsolicited events to request a service and to deliver responses. Elsewhere (column 7, lines 44-62) Aldred teaches three different types of port connections: event, command and null ports. The passage cited by the Examiner in the rejection of claim 6 clearly refers to issuing events via event ports. The cited passages do not mention any command port connection.

In the Examiner's Answer, the Examiner asserts that the parameters are submitted through the command port because "command ports allow the application to drive the receipt or supply of *data* to the port". The Examiner is clearly redefining parameters of data using hindsight speculation, which is improper. Furthermore, the Examiner cites columns 35 and 36 and asserts, "functions are submitted through the command port". The cited text does not disclose command ports at all; in fact, columns 35 and 36 repeatedly disclose events and event handlers associated with the functions that would be communicated through the event port. The cited section in column 24 clearly defines the process associated with the "appropriate parameters" through events. Events are passed along the event port. The Examiner is merely speculating that parameters are passed along the command port, but nowhere provides evidence or an example to support this.

Thus, the rejection of claim 6 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claim 17.

Third Ground of Rejection:

Claims 7 and 18 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Aldred, Raynak and Simonoff, and in further view of Jalili et al. (U.S. Patent 5,423,042) (hereinafter “Jalili”). Appellants respectfully traverse the rejection of claims 7 and 18 for at least the reasons presented above regarding their respective independent claims.

Further regarding claim 7, Appellants have argued that the Examiner’s combination of Aldred, Raynak, Simonoff and Jalili fails to teach or suggest passing a value of a memory location for storing results of a function trigger by the passing of the function value. The Examiner relies on Jalili, citing the Abstract and column 10, lines 33-48 of Jalili. However, Jalili does not teach passing a value of a memory location for storing results. Instead, Jalili teaches that a client uses a SET FUNCTION DATA request to supply the arguments for a requested function and a GET FUNCTION DATA to obtain the results of the function. When using both the SET FUNCTION DATA and the GET FUNCTION DATA, a client in Jalili’s system sends the server the function name, type and instance to identify the requested function and results (Jalili, column 9, lines 41-64). Jalili also teaches that the server, when executing a requested function, uses the values of the state 288 field of the exec_table. Jalili states, “[t]he results 289 value is a pointer to a server memory space where the results of running the function will be stored” (Jalili, column 10, lines 40-43). However, Jalili does not teach passing a value of a memory location for storing results. Instead, Jalili teaches that after performing the requested function, the results, “which are stored in the memory space pointed to by the results 289 value, are send back to the client” (Jalili, column 10, lines 43-48). Nowhere does Jalili mention passing a value of a memory location for storing results of a function triggered by the passing of the function value. Instead, as noted above, Jalili teaches passing the results of the executing a function.

In the Examiner's Answer, the Examiner refers to Jalili's teaching regarding the server passing the ITP to the entry function and Jalili's teachings describing that a function is passed the value in the state field 288 that points to pre-allocated space where the arguments to the function reside, citing column 7, lines 30-33 and column 9, lines 35-37. However, the Examiner has failed to consider the specification limitation of claim 7 regarding passing a value of a memory location for storing results of a function trigger *by passing the function value*. Nowhere does Jalili teach or suggest this limitation.

Appellants have further argued that Aldred, Raynak and Simonoff fail also fail to teach or suggest passing a value of a memory location for storing results of a function triggered by the passing of the function value and thus fail to overcome the above-noted deficiencies of Jalili. The Examiner combination of Aldred, Raynak, Simonoff and Jalili clearly fails to teach or suggest passing a value of a memory location for storing results of a function triggered by the passing of the function value.

Therefore, for at least the reasons presented above, the rejection of claim 7 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claim 18.

CONCLUSION

For the foregoing reasons submitted in the Appeal Brief and this Reply Brief, it is submitted that the Examiner's rejections of claims 1-20 was erroneous. Reversal of the Examiner's decision is respectfully requested.

The Commissioner is authorized to charge any fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-78901/RCK. This Reply Brief is submitted with a return receipt postcard.

Respectfully submitted,



Robert C. Kowert

Reg. No. 39,255

ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: June 30, 2006